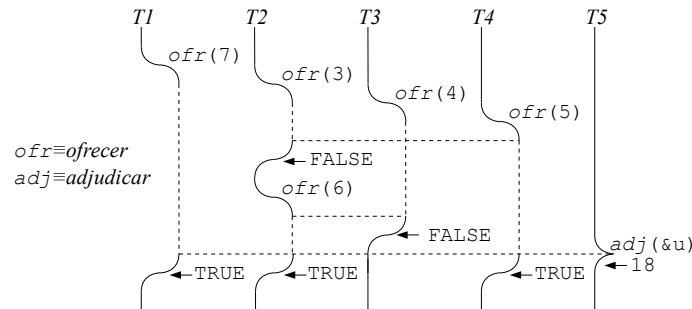


CC4302 Sistemas Operativos – Tarea 3 – Semestre Primavera 2023

En esta tarea Ud. deberá programar un sistema de subastas para threads. La API que Ud. debe implementar es la siguiente:

- *Subasta nuevaSubasta(int n)*: crea un nuevo sistema para subastar n unidades idénticas.
- *void destruirSubasta(Subasta s)*: libera los recursos de la subasta.
- *int adjudicar(Subasta s, int *prestantes)*: Cierra la subasta retornando el monto total recaudado y entregando en **prestantes* la cantidad de unidades sobrantes de la subasta, es decir las que no se vendieron porque llegaron menos que n oferentes.
- *int ofrecer(Subasta s, int precio)*: múltiples tareas invocan esta función para hacer una oferta por comprar una unidad del producto al valor *precio*. Esta función espera hasta que (i) la subasta se cierre, retornando TRUE, o (ii) n otras tareas ofrecieron un precio mayor por el producto, y en tal caso se retorna FALSE.

El siguiente diagrama de threads muestra un ejemplo de ejecución considerando $N=3$:



Observe que cuando *T4* ofrece 5, hay 4 oferentes siendo *T2* la peor oferta con *precio=3* y por lo tanto *T2* pierde retornando FALSE. Más tarde *T2* hace una contraoferta lo que transforma a *T3* en la peor oferta (*precio=4*) y retorna FALSE. Finalmente *T5* llama a *adjudicar* terminando la subasta. Los ganadores son *T1* que ofreció 7, *T2* con 6 y *T4* con 5, por lo que la recaudación asciende a 18. El valor final de *u* es 0 ya que se vendieron todas las unidades.

Requerimientos

- Ud. debe programar las funciones pedidas y la estructura del tipo *Subasta* en el archivo *subasta.c*.

- Ud. debe resolver esta tarea usando un mutex y múltiples condiciones como herramienta de sincronización. No puede usar semáforos.
- Use el patrón *request* eficientemente: Use múltiples condiciones para la espera en *ofrecer*. Debe evitar usar *pthread_cond_broadcast* para despertar a todos los threads que esperan en *ofrecer*. Use *pthread_cond_signal*.
- Use la cola de prioridades que está programada en *subasta.c*.

Instrucciones

Baje *t3.zip* de U-cursos y descomprímalo. El directorio *T3* contiene los archivos *test-subasta.c*, *Makefile*, *subasta.h* (con los encabezados requeridos) y otros archivos. Ud. debe programar en el archivo *subasta.c* las funciones solicitadas. Defina otras funciones si las necesita.

Pruebe su tarea bajo Debian 11 de 64 bits. Ejecute el comando *make* sin parámetros. Le mostrará las opciones que tiene para compilar su tarea. Estos son los requerimientos para aprobar su tarea:

- *make run* debe felicitarlo por aprobar este modo de ejecución.
- *make run-g* debe felicitarlo.
- *make run-san* debe felicitarlo y no reportar ningún incidente en el manejo de memoria.
- *make run-thr* debe felicitarlo y no reportar ningún datarace.

Cuando pruebe su tarea con *make run* en su computador asegúrese de que esté configurado en modo alto rendimiento y que no estén corriendo otros procesos intensivos en uso de CPU al mismo tiempo.

Invoque el comando *make zip* para ejecutar todos los tests y generar un archivo *subasta.zip* que contiene *subasta.c*, con su solución, y *resultados.txt*, con la salida de *make run*, *make run-g*, *make run-san* y *make run-thr*.

Entrega

Ud. solo debe entregar por medio de U-cursos el archivo *subasta.zip* generado por *make zip*. Recuerde descargar el archivo que subió, descargar nuevamente los archivos adjuntos y volver a probar la tarea tal cual como la subió a U-cursos. Solo así estará seguro de no haber entregado archivos incorrectos. Se descuenta medio punto por día de atraso. No se consideran los días de receso, sábados, domingos o festivos.